AUSTRALIAN OS9 NEWSLETTER

EDITOR: Gordon Bentzen   (07) 344-3881
SUB-EDITOR: Bob Devries   (07) 278-7209
TREASURER: Jean-Pierre Jacquet (07) 372-4675
LIBRARIAN: Jean-Pierre Jacquet (07) 372-4675
                 Fax Messages   (07) 372-8325
CONSULTANT: Don Berrie   (079) 75-3537
SUPPORT: Brisbane OS9 Users Group

CONTENTS

Boy! doesn't time fly when you are having fun, or when you have a lot of work to do. It seems only yesterday that we were putting together the last newsletter and I had some news about the CoCoFest held in Chicago USA early May.

I was indeed very lucky to have the opportunity to meet so many interesting people in Chicago and to make some new friends. Arrangements were made to exchange information and newsletters with a couple of the U.S. groups, and since our last newsletter we have received the Apr-May issue of "MOID", the newsletter of the U.S. OS-9 Users Group and also the May-Jun issue of "CoCo123", the publication of the Glenside Color Computer Club, Illinois.

MOID Article
A MOID article by Chris Perrault deals with two things. The first is an old thought revived, encouragement to upgraded OS-9 Level II to version 3.0. The second is to form a Level II Standards Committee. Chris correctly points out that Level II has matured since its release with many module patches, rewrites and general hacks which have resulted in a system which has really begun to let its true potential shine. All this however has the result that many different systems are in use with possible incompatibility and confusion to users and programmers. Chris goes on to suggest that a version 3.0 could be made from the original issue system with agreed standard patches etc. The end result would be a Level II version 3.0 OS-9 system which was a known quantity and which would perform predictably with any programme or utility.

So what are your thoughts? Should such a Version 3.0 upgrade be attempted? And if so, how are the "standards" to be selected? A standards committee??

We would be happy to pass on your thoughts and input.

Glenside CoCo123 Artilce
As one might expect, the May-June issue of this newsletter included a detailed CoCoFest report. In this report by Allen Huffman he praised the "International OS9 Consortium Discussion" which was organized by Peter Tutelaers, EUROS-9. I must confirm that this session was very well attended and a lot of interest shown in the concept of a "cooperation" in the exchange of information between the OS9 usergroups around the world.

I have just read a message which Peter left on Delphi for me dated June 15th. Ooops! sorry about that Peter :-). Anyway, like myself, Peter has obviously been busy since Chicago and things are now starting to move again.

I would be very grateful for any ideas or suggestions from our members which would support the exchange of OS-9 information between usergroups around the world. I would in turn pass your suggestions along to Peter Tutelaers in the Netherlands.

One of the very first questions to be addressed is that of; by what means is the data to be transferred? How is the material to be controlled? Who is responsible? Where does the "Cooperation" fit with existing usergroups and networks, such as the OCN (OS9 Community Network) which widely uses the Fido network for messages etc?

We really need to answer only a few questions, How? Where? When? and Why? - that should cover it.

FAILED TEST
Now I must admit to having failed the test. What am I on about? Well I received a letter today from the president of the U.S. OS-9 Usergroup in which he expressed his dissappointment at the lack of response from his attempt to initiate an exchange of news articles between user groups.

Jim DeStafeno mailed some of his newsletter articles on disk to Japan, Netherlands, Germany, Switzerland and Australia (c/o myself) late in May.

Some five weeks later, apparently NO replies. O.K. so I have failed the test, sorry Jim, I will fix.

We shouldn't have to rely on people like Jim to keep pushing things along, so I will do my bit.

I know that each of you can contribute something in the way of information, ideas, or questions. Please send these to us now!
                    Until next time, Cheers, Gordon.

Get Next Window
a programme from our PD archive

'Getnw' is a subroutine which is to be used with Ron Lammardo's 'shell+', v2.1. 'Shell+' is available in the database here. 'Getnw' is a subroutine which retrieves the name of the next available window and puts it into a 'shell+' subvariable.

I have my system set up to open two or three windows on startup. I like to run larger programs from shell scripts - especially those which diddle pallette colors. The script opens a window, starts the application, and closes the window when the application is done. This way, my original windows remain available while the application is running, and their colors don't get changed to some oddball combination. I keep the scripts in my execution directory with the executed attribute set.
Before 'shell+' and 'getnw', the limitation to this scheme was that each application had to be hardcoded to a specific window. If the '/w' wildcard window name were used in the script, the shell had no way of knowing which window was actually used, making it difficult to get rid of it, or to manipulate output to it from the script.

The format for 'getnw' is:

        getnw n

where n is an integer, 0 - 9. 'Getnw' will get the name of the next available window and store it in 'shell+' subvariable n. It can then be accessed by 'shell+' by using "%%n". (See 'shell+' documentation for details of this.) A bad value of n will just do nothing, without any error report. If another window is unavailable, a #221 error gets passed back to 'shell+', which will report it.

As an example, here is the script file I use to run wizpro.

    * script file to execute wizpro in next window
    getnw 0
    unlink getnw
    wcreate %%0 -s=1 0 0 40 24 2 4 4   .
    merge /dd/sys/stdfonts
    prompt Autolog file ([enter] = none) —
    var.1
    display 02 2d 2c 05 20 1b 21 </1 > %%0
    echo Starting Wizpro >%%0
    cx /dd/com/pro/cmds
    (wizpro %1 <>>%%0;deiniz %%0)&

The only potential problem here is if another process were to grab off the window inbetween the

'getnw' call and the 'wcreate' call — an unlikely possibility. If this were to happen, no damage would be done anyway — the 'wcreate' call would just result in an error and abort the script.

Enjoy, and if any questions, ask.

                            Dennis Skala

                    CIS: 73177,2365
                    Delphi: DENNYSKALA

Here's the source code (in Assembler for a change!)

```
*************************************************
*                                               *
* getnw - gets the name of the                  *
*         next available window                 *
*         into a shellplus subvariable          *
*                                               *
* Use:  getnw n —— where n= 0 to 9              *
*                  is the subvar into           *
*                  which the name is to be put  *
*                                               *
*         No error is returned for a bad        *
*         parameter, and no action is taken.    *
*         System Call errors are passed back    *
*         to shellplus.                         *
*                                               *
*                                               *
*         Copyright 1989 by Dennis Skala         *
*         All rights reserved. May be freely    *
*         distributed so long as this notice    *
*         is left intact.                       *
*                                               *
*************************************************

          nam getnw
          ifp1
          use /dd/defs/os9defs
          endc

shellsub  set $50
rev       set 1
attr      set reent+rev
type      set shellsub+objct

          mod len,name,type,attr,start,0
name      fcs /getnw/
          fcb rev
win       fcc "/w"
          fcb $0d cr

start     equ *
```

```
sl       lda ,x+ get char                          window
         cmpa #' leading space is OK                         bcc wl
         beq sl                                     cl       leay -1,y if error
         pshs a save char for now                            lda #$0d clear the entry
s2       lda ,x+ must not be another char before             sta ,y
<cr>                                                          bra errout and exit without error
         cmpa #' space is OK
         beq s2                                     wl       ldb #ss.devnm
         cmpa #$0d must be <cr> here                         tfr y,x point to subvar
         beq tl                                              os9 i$getstt get the name
         puls a clean stack                                  bcc rl
         bra errout and exit without error                  os9 i$close if error
                                                             bra cl
tl       puls a retrieve character                 rl       os9 i$close close the path
         suba #$30 ASCII to number                          bcs cl if error
         bmi errout invalid entry                  r2       lda ,y+ read the name
         cmpa #9                                             bpl r2 if not at end
         bhi errout invalid entry                           leay -1,y last char position
                                                             anda #%01111111 strip off hi bit
         ldb #81                                             sta ,y+ replace it
         mul calculate offset to variable area              lda #$0d
         leax d,u                                            sta ,y end with a <cr>
         lda #'/
         sta ,x+ name starts with /                errout   rts no action on error
         tfr x,y save the pointer for later                 emod
         lda #write.                               len      equ *
         leax win,pc                                        end
         os9 i$open open path to next available
```

ooooooooooOOOOOOOOOOooooooooooo

Floppy to Hard to Floppy
Backup Utility

FHF (Floppy to Hard to Floppy disk) Backup Utility

This utility was written because I hate to waste my time swapping floppy disks during a single drive Backup. Since I have a hard drive as well as a floppy drive, I decided to write a utility that writes the source floppy disk's entire contents to a hard disk file, prompts to swap in a formatted destination floppy disk, and writes the hard disk file out to the destination floppy disk.

By turning off the verify during the destination disk write and performing a verify pass afterward I was able to reduce the entire back up time to less than 7 minutes for a 720K disk on my system. Your time will vary depending on the speed of your hard and floppy disk I/O and the size of the floppy disk.

This program requires the 'SysCall' subroutine which comes with Basic09 in the Color Computer 3's Level 2 OS-9 package. The 'SysCall' subroutine must be in your current execution (CMDS) directory, or in memory. If you do not have the 'SysCall' subroutine, it is possible to edit the Basic09 source and remove the two lines that reference 'SysCall'. Save the edited source and then make a new packed version of FHF in your current exection (CMDS) directory. If you want to maintain the highest possible back up speed, you should manually turn off the write verify before running the modified FHF program. You should turn the write verify back on after the modified FHF program is finished.

Here are the instructions to get FHF ready to run:

1) Extract the packed Basic09 'Fhf' module from 'fhf.ar'.
2) Copy 'Fhf' to your current execution (CMDS) directory.
3) Copy 'RunB' to your current execution (CMDS) directory if it's not already there.
4) Copy 'SysCall' to your current execution (CMDS) directory if it's not already there.

Here are the instructions to run FHF:

1) Type 'Fhf' and press [ENTER].
2) FHF will ask you for the source/destination drive. Type in the floppy drive name (you must include the slash) and press [ENTER].
3) FHF will ask you for the temporary storage drive. Type in the hard drive name (you must include the slash) and press [ENTER].
4) FHF will print out the information you just gave, and ask you if it's correct. If the information is correct, type 'Y' or 'y' and press [ENTER]. If the information is incorrect, type 'N' or 'n' and press [ENTER]. If you wish to quit, type 'Q' or 'q' and press [ENTER].
5) If the information is correct, FHF will ask you to insert the destination disk into the floppy drive and press [ENTER] when ready. Once you do this, FHF checks the destination disk for size and ID.
6) FHF then asks you to insert the source disk into

the floppy drive and press [ENTER] when ready. Once you have done this, FHF checks the source disk's size. If the source and destination disks are not the same size, FHF will abort.
7) FHF copies the entire source disk to a temporary file on the hard drive. When that is done, FHF will ask you to insert the destination disk into the floppy drive and press [ENTER] when ready. If the destination disk is not the same as before, FHF will prompt you again.
8) Once FHF has copied the temporary hard disk file to the destination disk, it will delete the temporary file and verify the destination disk write. Any verify errors encountered will be reported to the standard error path.

I hope you find this utility useful! Bruce Isted CIS 76625,2273

Here's the sourcecode in Basic09:

```
PROCEDURE FHF
(* FHF (Floppy to Hard to Floppy disk) Backup Utility *)
(* written by Bruce Isted  CIS 76625,2273. *)
(* Released to the Public Domain 87/08/02 *)
(* for non-commercial use only. *)
BASE 0
DIM callcode,fd_path,hd_path,options(32),size_bytes(3):BYTE
DIM disk_id(2):INTEGER
DIM chunk_total,count,dest_size,error_count,src_size:REAL
DIM answer:STRING[1]
DIM fd_nam,hd_nam:STRING
DIM sector:STRING[256]
DIM chunk:STRING[8192]
TYPE registers=cc,a,b,dp:BYTE; x,y,u:INTEGER
DIM regs:registers
error_count:=0
REPEAT
PRINT
INPUT "Source/destination drive? (EG: /D0) ",fd_nam
INPUT "Temporary storage drive? (EG: /H0) ",hd_nam
PRINT
PRINT "Source/destination drive is "; fd_nam; "."
PRINT "Temporary storage drive is "; hd_nam; "."
INPUT "Is this correct? (Y/N/Q) ",answer
PRINT
UNTIL SUBSTR(answer,"YyQq")<>0
IF SUBSTR(answer,"Qq")<>0 THEN
PRINT
END
ENDIF
PRINT "Insert destination disk into "; fd_nam;
INPUT " and press [ENTER] when ready. ",answer
OPEN #fd_path,fd_nam+"@":READ
GET #fd_path,size_bytes
SEEK #fd_path,14.
```

```
GET #fd_path,disk_id(0)
dest_size:=size_bytes(0)*65536.+size_bytes(1)*256.+size_bytes(2)
CLOSE #fd_path
PRINT "Insert source disk into "; fd_nam;
INPUT " and press [ENTER] when ready. ",answer
OPEN #fd_path,fd_nam+"@":READ
GET #fd_path,size_bytes
src_size:=size_bytes(0)*65536.+size_bytes(1)*256.+size_bytes(2)
IF src_size<>dest_size THEN
PRINT
PRINT "FHF aborted:  source and destination disks don't match."
PRINT
END
ENDIF
CREATE #hd_path,hd_nam+fd_nam+".temp":UPDATE
SEEK #fd_path,.0
chunk_total:=INT(src_size/(SIZE(chunk)/256))
FOR count=1 TO chunk_total
GET #fd_path,chunk
PUT #hd_path,chunk
NEXT count
WHILE NOT(EOF(#fd_path)) DO
GET #fd_path,sector
PUT #hd_path,sector
ENDWHILE
CLOSE #fd_path
REPEAT
PRINT
PRINT "Insert destination disk into "; fd_nam;
INPUT " and press [ENTER] when ready. ",answer
PRINT
OPEN #fd_path,fd_nam+"@":READ
GET #fd_path,size_bytes
SEEK #fd_path,14.
GET #fd_path,disk_id(1)   .
CLOSE #fd_path
src_size:=size_bytes(0)*65536.+size_bytes(1)*256.+size_bytes(2)
UNTIL dest_size=src_size AND disk_id(0)=disk_id(1)
OPEN #fd_path,fd_nam+"@":UPDATE
callcode:=$8D
regs.a:=fd_path
regs.b:=$00
regs.x:=ADDR(options)
RUN syscall(callcode,regs)
options(8):=1
callcode:=$8E
regs.a:=fd_path
regs.b:=$00
regs.x:=ADDR(options)
RUN syscall(callcode,regs)
SEEK #hd_path,.0
FOR count=1 TO chunk_total
GET #hd_path,chunk
PUT #fd_path,chunk
NEXT count
WHILE NOT(EOF(#hd_path)) DO
GET #hd_path,sector
```

```
PUT #fd_path,sector
ENDWHILE
CLOSE #hd_path
DELETE hd_nam+fd_nam+".temp"
PRINT "Sectors copied:  "; src_size
PRINT "Verify pass... please wait."
ON ERROR GOTO 1000
SEEK #fd_path,.0
FOR count=0 TO src_size-1
GET #fd_path,sector
100
NEXT count
CLOSE #fd_path
PRINT "Verify errors:  "; error_count
PRINT
END
1000 PRINT #2,"Error #"; ERR; " at sector "; count
error_count:=error_count+1
GOTO 100
```

oooooooooooOOOOOOOOOOOooooooooooo

A Fern in C

You may remember, quite some time ago, we printed a few programmes written by Mr. Ted Martin, of which one was a programme to draw a fern on a graphics screen. He used a mathematical routine to plot the points on the fern pattern. Well, here it is again, only this time in C.

```
/*

PROGRAM TO GENERATE THREE DIMENSIONAL ITERATED
FUNCTION SYSTEMS. Printed in Fractal Programming
in C writen by Roger T. Stevens, M&T Publishing
Inc. Ported to the Color Computer by Marty Criswell

*/


#include <stdio.h> /* this header is in the DEFS dir    */
#include <math.h>  /* this header with the kreiderlib.1 */
#include <os9.h>   /* this header is in the DEFS dir    */
#include <buffs.h> /* this header is on multivue disk   */
#include <wind.h>  /* this header is on multivue disk   */
#include <lowio.h> /* for this file see our PD library  */

int LINEWIDTH,OPERATOR,XCENTER,YCENTER,ANGLE;
long PATTERN;
int adapt,mode;
int j, k, index, xscale,yscale,xoffset,yoffset,pr,p[4],pk[4];
int hues[8] = {2,10,11,14};
unsigned i;
float a[4],b[4],c[4],d[4],e[4],f[4],g[4],h[4],m[4],n[4],q[4],r[4],
ca,cb,cg,sa,sb,sg,x,y,z,newx,newy,alpha[4] = {30,45,15,95},
beta[4] = {115,105,70,40},gamma[4]={25,70,20,-30};
float rad_per_degree=0.0174533;
```

```
int path=STDOUT;
char ch;

main()

{
DWEnd(path);
DWSet(path,7,0,0,80,24,2,1,1);  /* window set up */
CurOff(path);
Palette(path,0,36);
Palette(path,2,7);
Palette(path,1,0);
Select(path);
a[0] = 0; a[1] = .83; a[2] = .22; a[3] = -.22;   /* fractal parameters */
b[0] = 0; b[1] = 0; b[2] = -.23; b[3] = .23;
c[0] = 0; c[1] = 0; c[2] = 0; c[3] = 0;
d[0] = 0; d[1] = 0; d[2] = .24; d[3] = .24;
e[0] = .18; e[1] = .86; e[2] = .22; e[3] = .22;
f[0] = 0; f[1] = .1; f[2] = 0; f[3] = 0;
g[0] = 0; g[1] = 0; g[2] = 0; g[3] = 0;
h[0] = 0; h[1] = -.12; h[2] = 0; h[3] = 0;
m[0] = 0; m[1] = .84; m[2] = .32; m[3] = .32;
n[0] = 0; n[1] = 0; n[2] = 0; n[3] = 0;
q[0] = 0; q[1] = 1.62; q[2] = .82; q[3] = .82;
r[0] = 0; r[1] = 0; r[2] = 0; r[3] = 0;
p[0] = 328; p[1] = 27879 ; p[2] = 30173; p[3] = 32767;
xscale = 40;
yscale = 33;
xoffset = 60;
yoffset = -125;
for (index=0; index<4; index++)
{
    ca = cos(alpha[index]*0.0174533);
    cb = cos(beta[index]*0.0174533);
    cg = cos(gamma[index]*0.0174533);
    sa = sin(alpha[index]*0.0174533);
    sb = sin(beta[index]*0.0174533);
    sg = sin(gamma[index]*0.0174533);
    index==1    index==3 ? FColor(path,0): FColor(path,2);
    image_draw();
}
fscanf(STDIN,"%c",&ch);          /* changed 'scanf' to 'fscanf' here. ED */
DWEnd(path);                     /* reset window */
DWSet(path,2,0,0,80,24,0,1,1);
Palette(path,0,63);
Palette(path,1,1);
}

image_draw()            /* calculate the points to draw */
{

int px,py;
float vx,vy;

x = 0;
y = 0;
z = 0;
```

```
for (i=1; i<=10000; i++)
{
   j = rand();
   k = (j < p[0]) ? 0 : ((j < p[1]) ? 1 : ((j < p[2]) ? 2 : 3));
   newx = (a[k]* x + b[k] * y + c[k] * z + n[k]);
   newy = (d[k] * x + e[k] * y + f[k] * z + q[k]);
   z = g[k] * x + h[k] * y + m[k] * z + r[k];
   x = newx;
   y = newy;
   vx = x*ca + y*cb + z*cg;
   px = vx*xscale + xoffset;
   vy = x*sa + y*sb + z*sg;
   py = (vy*yscale + yoffset);
   if ((px>=-320) && (px<320)&& (py>=-192) && (py<192)){
     plots (px,py,hues[index]);
   }
 }
}

plots(x, y, color)            /* put the point on the screen */
{

#define convert(x,y)    {x = (x + 319);   y = (88 - ((93*y) >> 7));}

Select(path);
convert(x,y);
Point(path,x,y);
}
```
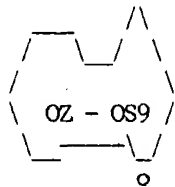
ooooooooooooOOOOOOOOOOOOooooooooooo

```
        __ /\
       /   \_/ \
      /         \
      \  OZ - OS9 /
       \___-\_/
          o
```

The National OS9 Usergroup
(07)-200-9870
300/1200/2400 baud.
20:00 to 22:30 HRS.(AEST)
(8N1)

Co-ordinator: Bob Devries (07)-278-7209
Sysop: Rod Holden

This is (RiBBS).... A Tandy Coco Based BBS program.
This BBS is accessable to Usergroup Members ONLY!
Feel free to look around , and test out the options.

OS9 for Ever !!!!

Hi, this is your Sysop once again letting you know what type of software is available. Here is the document on a programme called Lotto for all you people who need some help in selecting which numbers to

cross off.

LOTTO   OS9   LOTTO

NAME

Lotto - humorous lottery number picking game

SYNOPSIS

Lotto [-options]

DESCRIPTION

Lotto is a lotto number selection game. It is done in a somewhat satirical fashion somewhere between evangelism and random selection.

The '-t' option sets the top of the number range for selection. The default is 49.

The '-b' option sets the bottom of the number range for selection. The default is 1.

The '-n' option sets the number of numbers to be selected. The default is 6.

The '-a' option sets the number of 'real people testimonials' you wish to hear from previous lotto winners.

The '-?' and '-h' options both provide brief help messages.

EXAMPLES

Lotto

Plays lotto with defaults

Lotto -t99 -n7 -a3

Plays lotto with a top number of 99, picks 7 numbers, and gives 3 testimonials.

SEE ALSO

Psychic (This program is available also.)

Message Menu Area

Just a short note to let users know that when and if you decide to leave a message for other users, please use the Local message menu because the other message menus are for when we connect to outside BBS's. All you have to do is follow the prompts and you will be able to leave your message or messages.

If any user is running a BBS, could you please send me your details I will make up a list and place it on the OZ-OS9 BBS, remember, users only. See you in the bit stream, Happy CoCoing.

Sysop
Rod Holden

ooooooooooOOOOOOOOOOOoooooooooo

NATIONAL OS9 USERGROUP - AUSTRALIA
SUBSCRIPTION RENEWAL / APPLICATION

Subscription Renewal [__]   New Subscription [__]

Surname: _____   First Name :_____   Title : _____

Street: _____

Suburb: _____   State :_____   Postcode : _____

Country:_____

Home Phone: _____   Business Phone: _____

Age Group (please tick)   Under 18 [__]   18-25 [__]   26-35 [__]
                          36-45 [__]   46-55 [__]   over 55 [__]

Do you run OS9 Level 1[__]   OS9 Level 2[__]   OSK[__]   OS9000[__]

Type of Computer for OS9: _____   RAM _____K.

Diskette 5.25 Number [__]   Tracks [___]   Sides [__]

Diskette 3.5   Number [__]   720k [__]   1.4meg [__]

Hard Drive Meg [____]   Controller Type _____

Printer Type/Model _____

Modem Type _____

Special Interests _____

_____

Can you contribute articles to this Newsletter ? _____

Date : ____/____/____   Signature :_____

Amount Enclosed: $ _____   ($18-00 will cover you for 12 months)
                            (A$25-00 Overseas)
                  (CHEQUES PAYABLE TO: NATIONAL OS9 USERGROUP)

Please Return Completed Form to :-

NATIONAL OS9 USERGROUP   OR   NATIONAL OS9 USERGROUP
c/o GORDON BENTZEN   c/o J.P. JACQUET
8 ODIN STREET   27 HAMPTON STREET
SUNNYBANK QLD 4109   DURACK QLD 4077